2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Meta-Reinforcement Learning Based Cooperative Surface Inspection of 3D Uncertain Structures using Multi-robot Systems

Junfeng Chen[1], Yuan Gao[1], Junjie Hu[1], Fuqin Deng[1] and Tin Lun Lam[1,2,†]

***Abstract*— This paper presents a decentralized cooperative motion planning approach for surface inspection of 3D structures which includes uncertainties like size, number, shape, position, using multi-robot systems (MRS). Given that most of existing works mainly focus on surface inspection of single and fully known 3D structures, our motivation is two-fold: first, 3D structures separately distributed in 3D environments are complex, therefore the use of MRS intuitively can facilitate an inspection by fully taking advantage of sensors with different capabilities. Second, performing the aforementioned tasks when considering uncertainties is a complicated and time-consuming process because we need to explore, figure out the size and shape of 3D structures and then plan surface-inspection path. To overcome these challenges, we present a meta-learning approach that provides a decentralized planner for each robot to improve the exploration and surface inspection capabilities. The experimental results demonstrate our method can outperform other methods by approximately 10.5%-27% on success rate and 70%-75% on inspection speed.**

I. INTRODUCTION

The field of cooperative surface inspection investigates the problem of deploying a team of robots with sensors to cooperatively inspect the surface of 3D structures [1]. It plays a crucial role in many applications, such as surveillance in the disaster [2], automated patrols at sea under safety requirements [3], wind tunnel inspection [4], bridge surface reconstruction [5], and thus has attracted a notable volume of research over the past years. In these applications, a 3D model with a completely known or partially known geometric structure is to be inspected with prior knowledge in the form of meshes or grids. Although many of the existing works only aim to find a global path that can provide full coverage of the surface, there exist two disadvantages: i) most methods inherently assume that they only inspect the surface of a single complex structure by multi-robot systems (MRS) with a centralized control structure, thus ignoring potential tasks of inspecting multiple structures posing a huge challenge to computation power. ii) most works only focus on planning global structural coverage path with prior information about its size, shape, and position, thus they cannot potentially consider the structural uncertainties.

Early classic work in the task of surface inspection mainly designs a motion planning algorithm to provide a full coverage plan for non-planar surfaces [6], however, this work cannot consider exploiting the potential advantage of MRS, apparently leading to inefficient inspection. To overcome these issues, some approaches by using MRS with a centralized control structure to cooperatively inspect surface are presented in [1], [7]. The aim of these works is to efficiently inspect a complex structure, such as airplane model and bridge model by taking full use of a network of sensors. Unfortunately, when facing multiple structures randomly distributed on the ground shown in Fig. 1, it tends to be impractical and time-consuming for the aforementioned approaches to provide a global and optimal solution because searching for an efficient solution in an uncertain solver space is extremely difficult and requires a large computation power. In contrast to the centralized control of MRS, it is obvious that developing a decentralized MRS-based inspection system can take full advantage of different inspection capabilities, and the computational burden can be greatly reduced by simply making partial observations based on each robot's sensor.

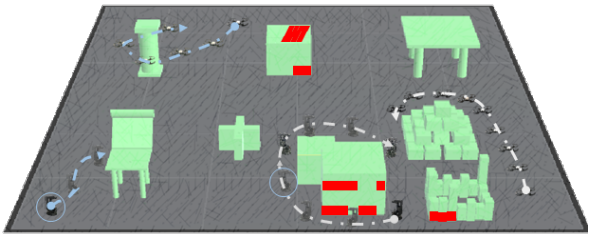


Fig. 1: An example of surface inspection in a 3D environment for MRS. In this scenario, the red area represents the inspected parts while the green area is the uninspected parts of structures.

It should be noted that most state-of-the-art approaches [4], [5] for planning cooperative coverage paths for a large and complex structure all require full prior knowledge of inspected structure, such as size, shape, number, and position. In practical environments, we argue that the most difficult challenge here is how to inspect multiple uncertain structures without any prior knowledge. Inspecting unknown structures is a complex task compared to tasks with completely known structures because there are three key steps to consider: i) exploring the whole environment to determine how many structures there are and where they are located; ii) figuring out their shape and size during inspection; iii) planning cooperative coverage path for each robot.

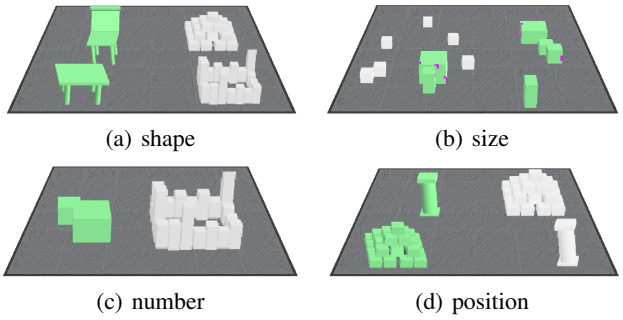
To address the two disadvantages mentioned above, in this paper, we present a meta multi-agent reinforcement

This paper is supported by the National Key R&D Program of China (2020YFB1313300) and Shenzhen Institute of Artificial Intelligence and Robotics for Society (AC01202101103).

[1]Authors are with the Shenzhen Institute of Artificial Intelligence and Robotics for Society

[2]Authors are with School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen

[†]Corresponding author: Tin Lun Lam tllam@cuhk.edu.cn



(a) shape (b) size

(c) number (d) position

Fig. 2: Examples of different environments. We use two kinds of colored structures, respectively green and gray to represent two kinds of states. We specifically consider four types of uncertainty factors, including shape, size, number, and position.

learning method-based multi-robot scanning system (*meta-MRSS*), that can perform inspection tasks without any prior knowledge. To be specific, we firstly sample the training data to form multiple tasks through the combination of different types of uncertainties. Then we apply meta multi-agent reinforcement learning methods to train an initial policy network for each robot to obtain general experience on uncertain structures. After that, the well-adapted initial policy network can be further trained in new uncertain structures. Finally, we can achieve fast inspection of uncertain structures with a small amount of training. Our method is tested on numerous types of uncertainties, including number, size, shape, and position of structures, as shown in Fig. 2. As a result, our method outperforms existing methods in terms of exploration efficiency and coverage effectiveness.

Our contributions are summarized as follows:

- We present a meta multi-agent reinforcement learning based method to handle the challenge when lacking any prior geometric information of structures. Our approach trains policies that can quickly adapt to uncertain structures and efficiently implement surface inspection.
- We propose a decentralized cooperative planning method for surface inspections of multiple structures by using MRS with different capabilities of sensors.
- Experimental results show that our method outperforms other methods including learning based method [8], [9] and NBVPlanner [10] by approximately 10.5%-27% on success rate and about 70%-75% on inspection speed.

II. Related Work

A. Surface Inspection of 3D Structures

In the literature on surface inspection, existing works strongly assume that the prior geometric information about structures' surface to be inspected can be easily provided manually or automatically as a mesh or grid model by relevant CAD software or previous mapping tasks [10]. One of the early works including [6] proposes a time-optimal approach for achieving complete coverage of 3D urban structure based on full prior features about abstract and simplified models capturing urban features. Alternatively, an iterative planning strategy is proposed to provide a complete global coverage path with the aid of re-meshing techniques in [11], and to improve computation efficiency, this work models viewpoints as Traveling Salesman Problem to output fast and feasible coverage path at each iteration. However, geometric information about structures in the above two papers are oversimplified, thus insufficient for planning complex and feasible coverage path with respect to large and complex structures. To properly overcome the aforementioned issue, a real-time coverage path replanning method is proposed by [12] for inspection of a large 3D underwater structure, with the assumption of a knowledge of a bathymetric map. It should be noted that using a single robot to inspect the surface of large and complex structures tends to be inefficient and time-consuming. Therefore, there is a growing number of approaches to improve inspection efficiency by planning cooperative inspection paths for MRS. In [1], the authors present a new and decentralized method for planning a cooperative and safe path for 3D surface surveillance by taking full account of the requirement of avoiding collision. Another alternation to solve the same problem of inspecting complex infrastructures is advanced by [4]. More specifically, by initially slicing the entire structures into specific branches and regions, then a cooperative coverage path planning approach with a centralized control structure is introduced to provide a separate path for every robot by solving a global optimization problem. In summary, aforementioned inspection algorithms proposed in the literature all assume a fully known environment.

In order to further study the problem of inspecting the surface of structures without any prior knowledge, a variant to the notion of the frontier which are the boundaries of inspected parts of surfaces from an unknown structure is proposed by [13]. This method allows the inspection of structural surface without acquiring any prior information. Similarly, a new path planning approach for robotic exploration and inspection is presented in [10], where the authors employ a variant of the next-best-view (NBV) planner [14] in a receding horizon fashion to firstly explore unknown structures, and then perform inspection task by exploiting just built occupancy map. However, the above two works only have a general assumption that they attempt to present path planning approaches for exploration and surface inspection with the use of a single robot or assign path by a centralized planner.

B. Meta-Reinforcement Learning

To adequately cope with the diversity and uncertainty of the environments, how to fast adapt into unknown environments outside training sets has aroused tremendous attention from researchers. It is widely believed that meta-learning [15]–[17] can be applied to quickly adapt to new environments while maintaining good performance in uncertain environments. A meta-learning method called MAML is proposed to quickly solve multi-tasks of cheetah locomotion in [18]. Inspired by this work, our approach is proposed to solve the problem of fast exploration and efficient surface inspection of MRS in the presence of multiple uncertain

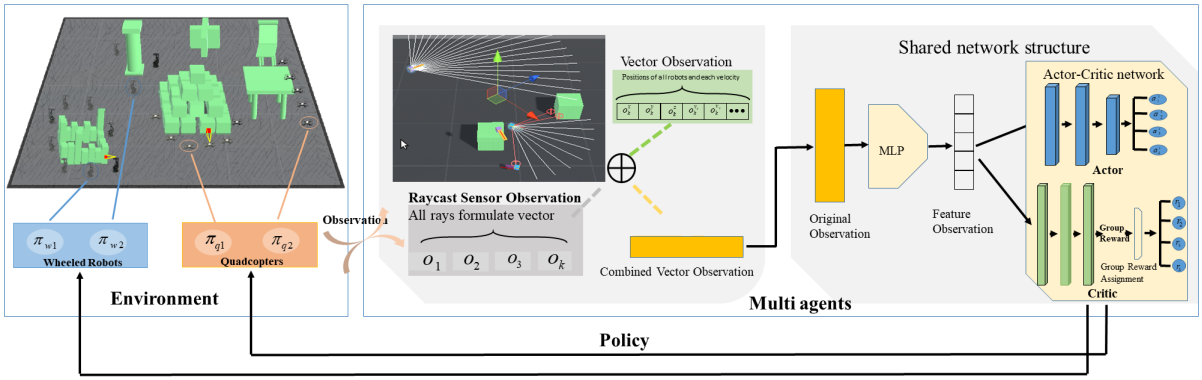


Fig. 3: The structure of meta optimizer. The wheeled robots and quadcopters use different sets of policy networks. They utilize decentralized actor and centralized critic networks. This method is well applicable to MRS.

3D structures. In [19], [20], a meta learning framework is proposed to extend into a diversity of new scenarios and applications, like earthquake rescue and traffic light control. In [21], meta-TD3 is presented to help UAVs quickly adapt to new target motion patterns and obtain better tracking effectiveness. In [22], the authors propose a Bayesian meta-learning method to quickly adapt to different robotics platforms. The key difference between our paper and others is that previous meta-reinforcement learning approaches are built on a single robot, while our method is built on MRS.

III. Methodology

A. Problem Formulation

We consider a group of robots $N_R = \{R_w \cup R_q\}$, in which wheeled robots represented by R_w can only move on the ground, while quadcopters labeled by R_q could fly in 3D space to inspect the entire 3D structures Φ_i around the whole environment Ω , namely $\Omega = \{\Phi_1, \Phi_2, \cdots, \Phi_n\}$ with uncertainty G. We assume that the uncertainty G as described in Fig. 2, refers to the position, number, shape and size of structures Φ_i. In our task settings, we only focus on inspecting 3D structures while ignoring the ground. As for each Φ_i, building on the surface model approach of [1], [10], we assume it consists of cells $\mathcal{C} = \{1, 2, \cdots, m\}$. which are essentially uniform grids by discretizing each structure Φ_i. It is clear that, according to the different sizes of each structure, these structures will have completely a different number of cells. The state of the cells is either uninspected (**0**) or inspected (**1**) ($\mathcal{C}_i \in \{0, 1\}$), according to whether the cell is perceived by sensors. As Fig. 1 shows, the small red grids essentially represent inspected cells, while green grids are regarded as uninspected cells. We assume that in these factors, positions and shape satisfy normal distribution while size and number obey geometric distribution. Robots N_R move around the environments with states $S_{t,R} \in \mathbb{R}^3$ at discrete time intervals via the dynamic models

$$S_{t,R} = f(S_{t-1,R}, U_{t,R}). \tag{1}$$

Specifical ly, the state of the wheeled robots R_w is $S_{t,R_w} = [x_{t,R_w}, y_{t,R_w}, 0]$. However, the state of the quadcopters R_q exists in 3D environments $S_{t,R_q} = [x_{t,R_q}, y_{t,R_q}, z_{t,R_q}]$. Note that although quadcopters can move around the whole environment, they are not allowed to fly close over the ground due to the effect of ground effect on smooth flying. Additionally, $U_{t,R} \in \mathcal{U}$ belongs to a finite set of control inputs, referring to velocity command in our settings. Because of safety requirements, MRS must remain a safety scanning region to avoid colliding with other robots or structures. Therefore, we should constrain the states of robots into a safety scale

$$S_{t,R} \in \mathcal{X}_{safe}(\Omega). \tag{2}$$

We consider that robots carry a sensor with fixed front orientation relative to the camera platform and a limited FoV of 20° with a shape of the square which has the same size as cells of structures Φ_i. The main difference of sensors between quadcopters and wheeled robots is that they have different sensing ranges, respectively 3.5 meters, and 2 meters. For a square facet on the surface to be considered visible by the sensors, it should satisfy the following conditions: i) its center is in the sensors' FoV; ii) its distance to sensors d is within a valid range; iii) its angle with respect to the sensors will be changed in our settings. Assuming that scanning measurement, including scanning range and viewpoint, is deterministic without adding Gaussian noise so that robots could determine definitely whether the cells of structures are inspected by thresholding effective scanning range via the object recognition method proposed by [23]. Therefore, when scanning the cells within effective sensor ranges, robots can infer the number of inspected cells $\Theta^{sensor}(S, \Omega) \subseteq \mathcal{C}$ which are rewarded as inspected values

$$y_{t,R} = h(S_{t,R}, \Phi) = \{c_i : i \in \Theta^{sensor}(S, \Omega)\}. \tag{3}$$

In Eq. (3), reward $y_{t,R}$ is proportional to $\Theta^{sensor}(S, \Omega)$. The objective is that MRS tries to inspect as many cells of uncertain structures as possible in a given time horizon:

$$\begin{aligned} \max_{\Omega \in G} \quad & \sum_{t' \in \{1,2,\cdots,t\}, R \in N_R} y_{t',R} \\ \text{s.t.} \quad & S_{t',R} \in \mathcal{X}_{safe}(\Omega_{t'}), \\ & S_{t',R} = f(S_{t',R}, U_{t',R}), \\ & y_{t',R} = h(S_{t',R}, \Theta), \\ & for\ all\ t' \in \{1, 2, \cdots, t\} and\ R \in N_R, \end{aligned} \tag{4}$$

Algorithm 1 meta-MRSS

Input: p(τ): samples from uncertainty distribution
Input: **I**: meta-update iterations
Output: θ_0: meta policy

1: randomly initialize $\theta_j \in \{\theta_1 \cup \theta_2\}$
2: **for** all iteration in **I do**
3: Sample buffers of tasks $\tau \sim p(\tau)$
4: **for** τ_i in τ **do**
5: Sample **K** trajectories $\mathbb{D}$ using f_{θ_j} in τ_i
6: Calculate loss $\mathcal{L}_{\tau_i}$ using $\mathbb{D}$ via meta optimizer in Eq.(6).
7: Compute adapted parameters by optimizer: $\theta'_{j,i}$
8: Sample validation trajectories respectively $\mathbb{D}'_{j,i}$ using $f_{\theta'_{j,i}}$ in τ_i
9: **end for**
10: Update θ_j via $\sum_{\tau_i \sim p(\tau)} \mathcal{L}_{\tau_i}(f_{\theta'_j})$ using $\mathcal{D}'_i$ and $\mathcal{L}_{\tau_i}$
11: **end for**

As the Eq. (4) describes, under satisfying the kinodynamic constraints f and collision avoidance $\mathcal{X}_{safe}$, we utilize MRS N_R to maximize the inspection efforts $y_{t',R}$ in terms of uncertain environments G.

B. meta-MRSS Algorithm

In this section, we will propose a meta multi-agent reinforcement learning method to solve optimization problem presented by Eq. (4). Firstly, due to the dependence of inspection actions of robots on current robot states, we model this optimization problem as Markov Decision Process (MDP). Then, in order to maximize the objective function represented by Eq. 3, we can regard the objective function as a reward that can be accumulated to obtain the maximum expected cumulative return by using multi-agent reinforcement learning algorithms. In addition, we can shape reward mechanisms to learn collision avoidance strategies. Finally, in terms of uncertain environment, we propose the meta-learning method to achieve good inspection performance in a short time horizon.

Our method as shown in Alg. 1 is divided into two parts, respectively meta optimizer in the inner loop and meta learner in the outer loop. For detail, refer to the work [18]. We only clarify required notations in Alg. 1, i.e., θ_1 and θ_2 respectively represent policy network parameters of wheeled robots and quadcopters, thus θ_j is joint policy's parameters needed to learn from above these robots. Throughout iterating over Alg. 1, an initial joint policy can be obtained labeled by θ_0, which can be flexibly extended into unseen tasks in the sequel.

Firstly, in the inner loop (as seen in lines 4-9), we make sure that the meta optimizer could cover in specific environments. Then we sample various training environments as multiple tasks, by utilizing meta learner we could obtain good-performance meta policy which can be used to learn a policy in uncertain environments with little training to realize fast adaptation.

1) Meta optimizer: To achieve this problem modeling, we build the model in terms of the aforementioned task in the simulation through the toolkit called *mlagents* [24]. As shown in the Environment part of Fig. 3, we discretize 3D structures into uniform green grids to simulate cells that will be inspected by sensors. If the cells are inspected by robots, they will immediately become red. To maximize optimization objective proposed by Eq. (4), we model this problem as a fully cooperative multi-agent task. It is generally accepted that MDP for multi-agent systems could be defined as: $\mathcal{M} = (\mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{O}, P, \Omega, r, \gamma)$, where $\mathcal{N} \equiv \{1, 2, \cdots, n\}$ is a set of agents and $\mathcal{S}$ is a finite set of global states. Due to system heterogeneity in that different types of robots have their own specific actions and states, we thus adopt a decentralized control strategy that we assign each policy network to each robot. At each time step, our system operates in a partially observable environment, in which each agent i receives a partial observation $o_i \in \Omega$ by following the observation probability function $\mathcal{O}(o_i|s, a_i)$. Every agent $i \in \mathcal{N}$ chooses the action $a_i \in \mathcal{A}$ based on a partial observation o_i, which results in a collective action space $\mathbf{a} \equiv \{a_1, a_2, \cdots, a_n\} \in \mathcal{A}$. Therefore we can easily obtain the robots' kinodynamic state presenting on the Eq. (1) by using *mlagents*. We seek the optimal joint policies $\pi^* = \{\pi_1, \pi_2, \cdots, \pi_n\}$ to maximize a joint value function $V^{\pi *}(\mathcal{O}) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, \pi^*]$, where $\mathbf{r}$ is the reward function and γ is discount factor. Besides MDP to be considered, communication mechanisms should also be most concerned for MRS. Specifically, we build the models in five aspects:

- *Joint action space*: To make robots' motion suitable for real-world applications, we set joint continuous action space. Specifically, we set forward-back, the left-right velocity of wheeled robots with $-0.2 \sim 0.2$ m/s and turn velocity of $-1 \sim 1$ rad/s. The main difference between quadcopters and wheeled robots is that forward-back, left-right, up-down velocity is $-0.3 \sim 0.3$ m/s. Besides, we also set the pitch and yaw velocity of quadcopters both being $-3 \sim 3$ rad/s.
- *Global observation space*: In the simulation, we deploy raycast sensors to collect observations. These raycast sensors emit rays every 5 degrees within a limited scanning range. When hitting the structures, rays will return the positions and state of cells. As shown in the Multi agents part of Fig. 3, to facilitate the learning process, we adopt one-hot ways to encode the state of the cells into 0 or 1 in our setting. Besides the observation of sensors, we also collect the positions and velocity as observation input.
- *Communication setting*: To achieve good-performance collaboration for MRS, we also consider the communication mechanism. We conduct some ablation experiments to demonstrate a good mechanism in which wheeled robots could access information of all robots while quadcopters only send the structure information to wheeled robots. Ablation study further shows that this mechanism well applies to the fact that quadcopters that have stronger motion capabilities can convey more useful information.
- *Reward shaping*: To avoid restricting solution space by

excessive prior experience, we adopt a relatively simple reward function. Overall, the reward function Eq. (5) is defined as follows:

$$\mathbf{r} = \begin{cases} sum(h(S_{t',R}, \Phi)) & \text{if robots inspect the cells} \\ -1 & \text{if robots collide with structures} \\ -1 & \text{if robots collide with other robots} \\ -0.05 & \text{time penalty at each step} \end{cases} \tag{5}$$

where we assume when wheeled robots or quadcopters inspect cells of these structures in a 3D environment they will be rewarded by $sum(\cdot)$, which refers to the addition of all elements in the tuple. To avoid obstacles including structures and other robots, if they collide, they will be punished at one point. To fully prove how the negative reward works for collision avoidance, we conduct the ablation study to analyze the effect. The reward scaling is from parameter tuning. Note that we set a time penalty at each time step so that MRS could finish the inspection task as quickly as possible. To solve the optimization problem, we encode the objective function as an accumulative reward, meaning that when we obtain the maximum accumulated reward, then we directly obtain the optimal solution of the Eq. (4).

- *Training method*: As shown in Fig. 3, we configure two sets of policy networks respectively for wheeled robots and quadcopters. Each network follows the actor-critic structure with central critics and distributed actors. We combine the observation vector of raycast sensors and another observation vector of robots' position and velocity as a combined observation vector. To reduce observation dimension and improve training efficiency, a 3-layer MLP is adopted to encode combined observation vector to output feature observation space, finally inputting it into the actor and critic network. The meta optimizer algorithm we are using is called MultiAgent POsthumous Credit Assignment(MA-POCA) [25]. Then we calculate the loss $\mathcal{L}_{\tau_i}$ advanced by Eq. (6). For each agent i, the advantage function that compares the Q-value for the current action a^i to counterfactual baselines that marginalize out a^i can be computed while keeping the other agents' actions $\mathcal{A}^{-i}$ are fixed.

$$\begin{aligned} \mathbb{A}^i(S, \mathcal{A}) &= Q(S, \mathcal{A}) - \sum_{a'^i} \pi^i(a'^i|\tau^i) Q(S, (\mathcal{A}^{-i}, a'^i)) \\ \mathcal{L}_{\tau_i}(f_{\theta_j,i}) &= -\mathbf{E}_{i_t \sim f_{\theta_j,i}}[\mathbb{A}^i(S, \mathcal{A})] \end{aligned} \tag{6}$$

where $\mathbb{A}^i(s, u)$ represents the counterfactual baseline of the single agent labeled by i, a^i is each agent's action, τ^i refers to the past action sequence. While $\mathcal{A}$ is the joint actions of all agents, $\mathcal{A}^{-i}$ presents the joint actions of other agents, S represents the global state. $Q(S, \mathcal{A})$ is used to estimate $Q-$values for the joint action $\mathcal{A}$ concerning the global state S.

2) Meta learner: As shown in Fig. 4, meta-MRSS utilizes MAML structure as meta learner (as shown in line 2-11 in Algorithm 1). In the inner loop, in order to simulate uncertain environments, we randomly sample a large number of different tasks through domain randomization mechanism conditioned on different uncertainty factors. As for each

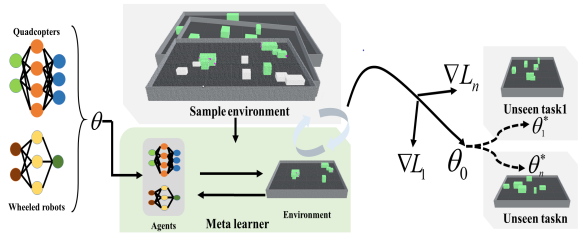


Fig. 4: The structure of meta-MRSS. Firstly, meta policies θ_0 are obtained through meta-MRSS. Then the optimal policies θ_i^* for uncertain structures could be learned through little training of meta policies.

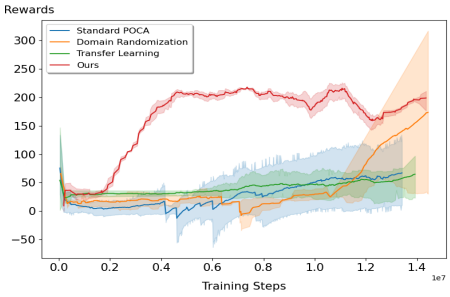
training task, we utilize the copies of meta policy to train and compute loss. In addition, we use more than one gradient descent update due to the dynamic and complexity of MRS. In the outer loop, we further compute the average total loss to finally update meta policy θ_j given that the inner loop provides sufficient loss through training on the buffers of tasks. The illustration is shown in Algorithm 1. The advantage of our proposed algorithm is that when updated meta policies are applied to uncertain tasks, we only train on a few iterations to obtain optimal policies on new tasks in order to achieve fast inspection for uncertain structures.

IV. Experiments

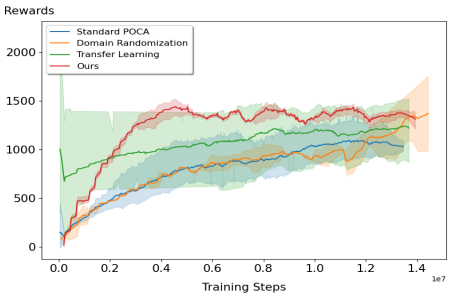
1) Experimental Setup: As shown in Fig. 2, to fully simulate the uncertain environments, we randomly sample 1000 environments as a training set, then we sample 5 unknown environments as the test set to verify the performance of our method in the test set. To make a fair comparison, we train these networks under the same hyperparameters by setting different stochastic seeds to avoid experimental serendipity and provide distributed policy network for each robot.

2) Baseline Methods: To quantify our method, we provide a thorough evaluation of our method by comparing it with existing methods. Specifically, we choose the following four methods for dealing with the uncertainties as a baseline:

- **Standard POCA** Since our method is built on POCA, therefore we conduct experiments to fairly compare our method with standard POCA. We utilize standard POCA to obtain the average rewards in the test set.
- **Domain Randomization** Domain Randomization mechanism [8] is a common way to improve the general capability of reinforcement learning. Therefore, we simulate these uncertainties through *mlagents*, then use the POCA algorithm in these stochastic simulation environments to obtain a model with high generalization capability.
- **Transfer Learning** Considering that most tasks are relevant, transfer learning [9] allows sharing the learned model parameters for new models to accelerate and optimize the learning efficiency. We first obtain the initial networks on the training set, and then perform a second training on the test set to obtain the average rewards.
- **Receding Horizon Path Planner (NBVPlanner)** NBV-Planner [10] is presented to provide a global coverage path for robotic exploration and inspection. This method plans the path in a geometric random tree, which can be regarded as a



(a) The learning process of the wheeled robots.



(b) The learning process of the quadcopters.

Fig. 5: The comparison results of learning for two kinds of robots using respectively four methods. The results present meta-MRSS has better learning efficiency in the face of uncertain structures.

variant of the rapidly exploring random tree (RRT) method. Therefore, we can design an objective function enabling the plan for inspection of the given surface in unknown structures. Therefore, we utilize NBVPlanner in the test set to calculate the average success rate of the inspection for unknown structures.

3) Experimental Results: To compare the performance of these baseline methods with meta-MRSS, we use the average reward and inspection success rate in the test set through three trials as metrics. The evaluation results are respectively shown in Fig. 5. Note that NBVPlanner is essentially an optimization solver rather than a learning-based method, therefore we compare success rate instead of average reward of this planner. As indicated by the results in Fig. 5 (a) & (b), our method can converge in nearly 4 million steps, while other learning-based methods require more than 14 million steps to reach convergence. Statistically, our method can outperform other baseline methods in terms of inspection speed by approximately 70%-75%. This proves that our method quickly provides better cooperative policies for MRS to inspect uncertain structures. From the result, we can conclude that, compared with these learning-based methods, our proposed method can learn a high generalization capability of initial policy by unifying different uncertainty factors, and then the initial policies can be quickly adapted to another optimal policy for uncertain structures with only little training process when facing uncertain structures.

In terms of inspection success rate, Fig. 6 turns out that our method can achieve a better inspection performance in the test set. Statistically, our method can reach an average success rate approximately of 87.12%, Domain Randomization-based POCA can reach 68.54%, Standard POCA's success rate is 60.16% and Transfer Learning can reach 71.74%. We observe that although Transfer Learning has a higher inspection success rate than other methods at the beginning when facing uncertain environments, it needs time to fine-tune the networks which can lead to a steep drop in inspection efforts. It is obvious that the curves labeled by Standard POCA slowly increase because POCA has weak generality in uncertain environments. Domain randomization-based method efficiently improves the generality of POCA, however, it has a lower level of inspection success rate by comparing with our method in given steps. In addition to comparing with learning-based methods, we further choose NBVPlanner as a non-learning-based method to demonstrate the performance of our proposed method. It can be seen that in Fig. 6 the success rate of initial iterations is close to 0. This is because NBVPlanner needs to firstly construct the occupancy map and then plan the global coverage path for surface inspection. For NBVPlanner the scenarios from the test set are executed 3 times, as the outcome is stochastic due to the use of RRT algorithms. The final results show the average success rate using NBVPlanner reaches 76.6%, which is inferior to our proposed method. This result is basically caused by the fact that our proposed method obtains general initial experience in inspecting uncertain structures so that initial policies can quickly adapt to other uncertain structures. However, NBVPlanner tends to build a map of uncertain structures by iterating many times before the inspection. Therefore, our proposed method has a better inspection success rate for a given time horizon.

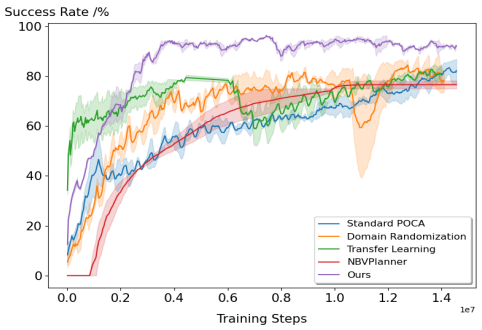


Fig. 6: The success rate of inspection using five methods.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we study the problem of how MRS could collaborate on the task of surface inspection of 3D uncertain structures. We attempt to use a decentralized method based on meta multi-agent reinforcement learning to deal with the uncertainties that appear in unknown environments. Experimental results demonstrate that our method outperforms other state-of-the-art approaches in terms of efficiency and effectiveness of surface inspection. Future work considers using above methods in the real multi-robot systems for real surface inspection application.

REFERENCES

[1] A. Adaldo, S. S. Mansouri, C. Kanellakis, D. V. Dimarogonas, K. H. Johansson, and G. Nikolakopoulos, "Cooperative coverage for surveillance of 3d structures," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1838–1845.

[2] W. Luo and K. Sycara, "Adaptive sampling and online learning in multi-robot sensor coverage with mixture of gaussian processes," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6359–6364.

[3] D. Guo, Y. Bai, M. Svinin, and E. Magid, "Robust adaptive multi-agent coverage control for flood monitoring," in *2021 International Siberian Conference on Control and Communications (SIBCON)*. IEEE, 2021, pp. 1–5.

[4] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, "Cooperative coverage path planning for visual inspection," *Control Engineering Practice*, vol. 74, pp. 118–131, 2018.

[5] R. Almadhoun, T. Taha, L. Seneviratne, and Y. Zweiri, "Multi-robot hybrid coverage path planning for 3d reconstruction of large structures," *IEEE Access*, vol. 10, pp. 2037–2050, 2021.

[6] P. Cheng, J. Keller, and V. Kumar, "Time-optimal uav trajectory planning for 3d urban structure coverage," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 2750–2757.

[7] H. Zhu, J. J. Chung, N. R. Lawrance, R. Siegwart, and J. Alonso-Mora, "Online informative path planning for active information gathering of a 3d surface," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1488–1494.

[8] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[9] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.

[10] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon path planning for 3d exploration and surface inspection," *Autonomous Robots*, vol. 42, no. 2, pp. 291–306, 2018.

[11] K. Alexis, C. Papachristos, R. Siegwart, and A. Tzes, "Uniform coverage structural inspection path–planning for micro aerial vehicles," in *2015 IEEE international symposium on intelligent control (ISIC)*. IEEE, 2015, pp. 59–64.

[12] E. Galceran, R. Campos, N. Palomeras, M. Carreras, and P. Ridao, "Coverage path planning with realtime replanning for inspection of 3d underwater structures," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6586–6591.

[13] L. Yoder and S. Scherer, "Autonomous exploration for infrastructure modeling with a micro aerial vehicle," in *Field and service robotics*. Springer, 2016, pp. 427–440.

[14] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon" next-best-view" planner for 3d exploration," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1462–1468.

[15] C. Lemke, M. Budka, and B. Gabrys, "Metalearning: a survey of trends and technologies," *Artificial intelligence review*, vol. 44, no. 1, pp. 117–130, 2015.

[16] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, "Meta-learning in neural networks: A survey," *CoRR*, vol. abs/2004.05439, 2020. [Online]. Available: https://arxiv.org/abs/2004.05439

[17] G. Papoudakis, F. Christianos, A. Rahman, and S. V. Albrecht, "Dealing with non-stationarity in multi-agent deep reinforcement learning," 2019. [Online]. Available: https://arxiv.org/abs/1906.04737

[18] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1126–1135.

[19] H. Jia, B. Ding, H. Wang, X. Gong, and X. Zhou, "Fast adaptation via meta learning in multi-agent cooperative tasks," in *2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 2019, pp. 707–714.

[20] S. Yang and B. Yang, "A meta multi-agent reinforcement learning algorithm for multi-intersection traffic signal control," in *2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2021, pp. 18–25.

[21] B. Li, Z. Gan, D. Chen, and D. Sergey Aleksandrovich, "Uav maneuvering target tracking in uncertain environments based on deep reinforcement learning and meta-learning," *Remote Sensing*, vol. 12, no. 22, p. 3789, 2020.

[22] A. Ghadirzadeh, X. Chen, P. Poklukar, C. Finn, M. Björkman, and D. Kragic, "Bayesian meta-learning for few-shot policy adaptation across robotic platforms," 2021.

[23] P. Carbonetto, N. De Freitas, and K. Barnard, "A statistical model for general contextual object recognition," in *Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I 8*. Springer, 2004, pp. 350–362.

[24] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, and D. Lange, "Unity: A general platform for intelligent agents," 2020.

[25] A. Cohen, E. Teng, V. Berges, R. Dong, H. Henry, M. Mattar, A. Zook, and S. Ganguly, "On the use and misuse of absorbing states in multi-agent reinforcement learning," *CoRR*, vol. abs/2111.05992, 2021. [Online]. Available: https://arxiv.org/abs/2111.05992